

Express Mail Label #: EJ721789044US

SPECIFICATION

IBM Docket No. STL920000091US1

TO ALL WHOM IT MAY CONCERN:

BE IT KNOWN that We, Howard J. Glaser of San Jose, California and citizen of the United States, Laurence E. England of Morgan Hill, California and citizen of the United States, Rebecca Lau Poole of San Jose, California and citizen of the United States, and Chenhong Xia of San Jose, California and resident of the United States, have invented new and useful improvements in

**METHOD, SYSTEM, COMPUTER PROGRAM PRODUCT, AND ARTICLE OF
MANUFACTURE FOR DOWNLOADING A REMOTE COMPUTER PROGRAM
ACCORDING TO A STORED CONFIGURATION**

of which the following is a specification:

002207 "4"42350

1
2
3 **METHOD, SYSTEM, COMPUTER PROGRAM PRODUCT, AND ARTICLE OF**
4 **MANUFACTURE FOR DOWNLOADING A REMOTE COMPUTER PROGRAM**
5 **ACCORDING TO A STORED CONFIGURATION**
6
7
8

9 **CROSS-REFERENCE TO RELATED APPLICATIONS**

10 *Sub A7*
11 Application Serial Number _____, filed concurrently herewith on October 12,
12 2000 for METHOD, SYSTEM, COMPUTER PROGRAM PRODUCT, AND ARTICLE OF
13 MANUFACTURE FOR INSTALLATION AND CONFIGURATION OF A COMPUTER
14 PROGRAM ACCORDING TO A STORED CONFIGURATION (IBM Docket
15 STL920000062), currently co-pending, and assigned to the same assignee as the present
16 invention; and

17 Application Serial Number _____, filed concurrently herewith on October 12,
18 2000 for METHOD, SYSTEM, COMPUTER PROGRAM PRODUCT, AND ARTICLE OF
19 MANUFACTURE FOR UPDATING A COMPUTER PROGRAM ACCORDING TO A
20 STORED CONFIGURATION (IBM Docket STL920000092), currently co-pending, and
21 assigned to the same assignee as the present invention.

22 The foregoing co-pending applications are incorporated herein by reference.
23
24

25 A portion of the Disclosure of this patent document contains material which is subject
26 to copyright protection. The copyright owner has no objection to the facsimile reproduction by
27 anyone of the patent document or the patent disclosure, as it appears in the Patent and
28 Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates in general to computer programs, and more particularly to a downloading of a computer program.

2. Description of the Related Art

Customers and providers may have a number of requirements in installing or deploying an application program.

Customers may prefer to remotely install application program clients from a central server(s) in order to increase both installation productivity and maintenance productivity. Thin clients may be preferred as thin clients generally require less workstation customization, perhaps as little as a browser with the appropriate option settings and plugins. Preferably, users should be able to move between different workstations, and still be able to access their tools and data, another potential advantage of a thin client. A customer may prefer to control which application programs and which versions of the application programs are available to a particular user. In addition to controlling the application programs, a customer may wish to control access to resources and assets based upon a user's responsibilities and roles in an organization. Alternatively, the customer may wish to control and filter project work views based on the user's responsibilities and roles. In addition to such central control, the customer may want an inventory of the application programs (including versions) installed and used on a workstation or a group of workstations without the need to go to each user workstation.

1 An application program provider may also have a number of similar requirements in
2 installing or deploying an application program. The provider may also want for licensing
3 purposes the inventory of the application programs (including versions) installed and used on a
4 workstation or a group of workstations without the need to go to each user workstation. To
5 satisfy a customer's specific individual requirements, the provider would prefer a mechanism
6 for configuring the application program to support a customer solution which may be tailored
7 by the customer to suit the customers needs, such as role-based views.

8
9 Conventional systems and methods have failed to provide all of the above advantages.
10 Thus, there is a clearly felt need for a method, system, article of manufacture, and computer
11 program product for providing application program downloading with these advantages.

SUMMARY OF THE INVENTION

The present invention comprises a method, system, article of manufacture, and a computer program product for downloading an application program. A configuration of the application program corresponding to a particular user of the local application is defined and stored on a remote server. In response to the user requesting the application program, a session is initiated between the local data processing system and the remote data processing system, and the user is authenticated. Responsive to the user authentication, data is downloaded from the remote server to the local data processing system according to the stored configuration. This stored configuration also may be encrypted and stored in a manifest file which is decrypted to produce a decrypted configuration in response to the user authentication. Data may then be downloaded from the remote data processing system to the local data processing system according to the decrypted configuration. The application program may be constructed pursuant to the configuration file or the decrypted configuration file.

One aspect of a preferred embodiment of the present invention defines a configuration of an application program corresponding to a particular user of the application program.

Another aspect of a preferred embodiment of the present invention stores the configuration on a remote server executing on a remote data processing system.

Another aspect of a preferred embodiment of the present invention initiates a session between the local data processing system and the remote data processing system in response to the user requesting the application program.

Another aspect of a preferred embodiment of the present invention authenticates the user in response to the user request.

Another aspect of a preferred embodiment of the present invention determines that the

1 stored configuration corresponds to the requesting user.

2
3 Another aspect of a preferred embodiment of the present invention downloads data
4 from the remote data processing system to the local data processing system according to the
5 stored configuration.

6
7 Another aspect of a preferred embodiment of the present invention encrypts the stored
8 configuration into a manifest file.

9
10 Another aspect of a preferred embodiment of the present invention downloads the
11 manifest file from the remote data processing system to the local data processing system.

12
13 Another aspect of a preferred embodiment of the present invention decrypts the
14 manifest file in response to the user authentication.

15
16 Another aspect of a preferred embodiment of the present invention builds the
17 application program pursuant to the configuration decrypted from the manifest file.

18
19 Another aspect of a preferred embodiment of the present invention authenticates the
20 user in response to the user requesting a build of the application program.

21
22 The present invention has the advantage of providing improved downloading of a
23 computer program.

24
25 The present invention has the further advantage of providing a download of an
26 application program client from a central server(s) in order to increase installation productivity
27 and maintenance productivity.

28
29 The present invention has the further advantage of providing control over which

1
2
3
4
5
6
7
8
9
0
1
2
3
4
5
6
7

00012000 17
" 4444 16
111111 15
222222 14
333333 13
444444 12

Figure 3 and **Figure 4** are flowcharts of method steps preferred in carrying out a preferred embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENT

An embodiment of the invention is now described with reference to the figures where like reference numbers indicate identical or functionally similar elements. Also in the figures, the left most digit of each reference number corresponds to the figure in which the reference number is first used. While specific configurations and arrangements are discussed, it should be understood that this is done for illustrative purposes only. A person skilled in the relevant art will recognize that other configurations and arrangements can be used without departing from the spirit and scope of the invention. It will be apparent to a person skilled in the relevant art that this invention can also be employed in a variety of other devices and applications.

With reference now to the figures, and in particular with reference to **Fig. 1**, there is depicted a pictorial representation of a distributed computer system **8** which may be utilized to implement the method of, system for, article of manufacture, and computer program product of the present invention. As may be seen, distributed computer system **8** may include a plurality of networks **10** and **32**, which may be Local Area Networks (LAN), intranet networks, or internet networks, each of which preferably includes a plurality of individual computers **12** and **30**, respectively. Of course, those skilled in the art will appreciate that a plurality of Intelligent Work Stations (IWS) coupled to a host processor may be utilized for each such network.

As is common in such data processing systems, each individual computer may be coupled to a storage device **14** and/or a printer/output device **16**. One or more such storage devices **14** may be utilized, in accordance with the present invention, to store the various computer programs which may be accessed and executed by a user within the distributed computer system **8**, in accordance with the present invention. In a manner well known in the prior art, each such computer program may be stored within a storage device **14**.

Still referring to **Fig. 1**, it may be seen that distributed computer system **8** may also include multiple mainframe computers, such as mainframe computer **18**, which may be

preferably coupled to Local Area Network **10** by means of communication link **22**. Mainframe computer **18** may also be coupled to a storage device **20** which may serve as remote storage for Local Area Network **10** which may be coupled via communications controller **26** and communications link **34** to a gateway server **28**. Gateway server **28** is preferably an individual computer or Intelligent Work Station which serves to link Local Area Network **32** to Local Area Network **10**.

As discussed above with respect to Local Area Network **32** and Local Area Network **10**, a plurality of server computer programs may be stored within storage device **20** and executed by mainframe computer **18**. Similarly, a plurality of client computer programs may be stored within storage devices **14** and executed by individual computers **12** such that distributed client/server computer programs are provided. Of course, those skilled in the art will appreciate that the mainframe computer **18** may be located a great geographical distance from Local Area Network **10**, and similarly, Local Area Network **10** may be located a substantial distance from Local Area Network **32**. That is, Local Area Network **32** may be located in California while Local Area Network **10** may be located within Texas and mainframe computer **18** may be located in New York.

As will be appreciated upon reference to the foregoing, it is desirable for a user of a workstation **12** to be able to remotely install an application program from server **18**. A configuration of the application program corresponding to this particular user of the application program is defined and stored on a remote storage **20** of the remote server **18**. In response to a user request communicated from the user workstation **12** to the server **18** via LAN **10** and communications link **22**, the server **18** determines that the configuration stored on storage **20** corresponds to the requesting user. Server **18** authenticates the user, and after such authentication, data is downloaded to the workstation **12** from the remote server **18** according to the stored configuration. This stored configuration is also encrypted and stored in a manifest file which is also downloaded. Responsive to the user authentication, the manifest file is decrypted, and the application program is constructed, configured, and installed on workstation

1 12 pursuant to the configuration decrypted from the manifest file. The user may also move
2 from a first workstation 12 on which the application program is installed to a different second
3 workstation 12 on LAN 10 and still be able to access the user's tools and data which may be
4 downloaded and installed on the second workstation 12 in a similar fashion to the download
5 and installation on the first workstation 12. Using the user configurations and authentications
6 stored on storage 20 and the server 18, a customer may control the application programs, the
7 versions of these application programs, and resources that are available to the particular user.
8

9 The customer may also inventory the application programs (including versions)
10 installed and used on a workstation 12 or a group of workstations 12 on LAN 10 without the
11 need to go to each user workstation. An application program provider may also want for
12 licensing purposes such an inventory which may be produced by customer's server 18 or the
13 provider's server 26.
14
15

16 Referring now to **Figure 2**, a block diagram of a preferred embodiment of the present
17 invention is illustrated. A file known as a manifest file or manifest 244 provides a description
18 as to which plugins, configurations, and other resources should be loaded when an application
19 program, such as an Integrated Development Environment (IDE), is initialized. Although the
20 preferred embodiment is presented in the context of an IDE, those skilled in the art will
21 recognize that the invention may be practiced with other types of application programs. The
22 preferred embodiment uses this manifest to provide a mechanism for plugging in tools or
23 components to provide a customized application program with the appropriate viewers, editors,
24 browsers, and other resources to support the user's needs. For example, a customized IDE
25 may be provided with the appropriate viewers, editors, and browsers to support a specific
26 developer's needs. Furthermore, the preferred embodiment allows remote customization and
27 configuration of the application program for an individual user based on the user's role and
28 responsibility in an organization.
29

1 The preferred embodiment of the present invention comprises an application server
2 **202**, which may be behind a web server, that is used to define users **204**, **206**, and **208** in a user
3 table **210** and to define the users' application program configurations (**214**, **216**, and **218**) in a
4 user IDE configuration table **220**. This configuration may include plugins **222** to be
5 downloaded , user authorizations for each user **224**, templates for builds **226** (pre packaged
6 site JCL), location/configuration of site servers **228**, other information **230** such as the location
7 and configuration of system input/output (I/O), and any site specific documentation **232** which
8 may be shown in a browser or plugin.

9
10 A security system **234**, such as a certificate or simple a logon dialog, may be used by a
11 user to initiate a session **236** with the server **202**. Upon authentication of the user by the
12 security system **234**, a table lookup of the user table **210** is performed and plugins **222**,
13 authorizations **224**, templates **226**, server information **228**, and documentation **232** are
14 aggregated into the user's components **238** which are downloaded via the download **240** to the
15 user's workstation **242**. In addition, a manifest **244** is created with this configuration
16 information. This manifest **244** is encrypted and requires authentication from a security system
17 **246** on the workstation **242** to be decrypted and used to build **248** an application program **250**.

18
19 After the download **240** to the user's workstation **242**, the user through a local logon in
20 the security system **246** invokes the decryption and use of the manifest **244** to build **248** the
21 user's personalized application program **250**. Meanwhile a request may be sent, preferably
22 unknown to the user, to the server **202** to check a component configuration table **252**
23 maintained in the application server **202** to determine if any items have been updated.
24 Alternatively, a request may be sent to the server **202** to record in a components inventory **254**
25 which tools are being used by the user.

26
27 Referring now to **Figures 3** and **4**, the flowcharts **300** and **400** illustrate the operations
28 preferred in carrying out the preferred embodiment of the present invention. In the flowcharts,
29 the graphical conventions of a diamond for a test or decision and a rectangle for a process or

function are used. These conventions are well understood by those skilled in the art, and the flowcharts are sufficient to enable one of ordinary skill to write code in any suitable computer programming language.

After the start **305** of the process **300**, process block **310** defines a configuration of the application corresponding to a particular user **206** of the application, such as user 2, and process block **315** then stores the configuration **216** on a remote server **202** executing on a remote data processing system **18**. Alternatively, the configuration **216** may be stored on local data processing system **242**. Thereafter, process block **320** may initiate a session between the local data processing system **242** and the application server **202** on remote data processing system **18** in response to the user **206** requesting the application. Responsive to this user request, process block **325** authenticates the user **206**, and process block **330** then determines that the stored configuration **216** corresponds to the requesting user **206**. Process block **335** then downloads data **240** from the remote data processing system **202** to the local data processing system **242** according to the stored configuration **216**. Alternatively, process block **335** may access the data **240** from the local data processing system **242** or from another resource within the distributed computer system **8** according to the stored configuration **216**. Thereafter, processing continues to decision block **450** on **Figure 4**. This processing path is illustrated by flowchart connectors **A**, **340** on **Figure 3** and **445** on **Figure 4**. Process block **450** encrypts the stored configuration **216** into a manifest file **244** which is then downloaded by process block **455** from the remote data processing system **202** to the local data processing system **242**. Alternatively, process block **450** may encrypt the stored configuration **216** into a manifest file **244** which is then stored on the local data processing system **242**. After the download **240** of the manifest file **244** and the components **238** to the user workstation **242**, the user **206** may request a build **248** of the downloaded components **238** pursuant to the configuration in the manifest file **244**. Responsive to such a user request, the user is authenticated by process block **460**, and if the user **206** is authenticated, then process block **465** decrypts the manifest file **244**. Thereafter, process block **470** builds, installs, and configures the application program **250** pursuant to the configuration decrypted from the manifest file **244**.

1 The process then ends at process block **475**.

2
3 In an alternative embodiment of the present invention, an application program is
4 updated for execution on the data processing system. In this alternative embodiment, process
5 block **330** also determines if any items described in the configuration have been updated. If an
6 item has been updated, then process block **335** retrieves the updated item, and process block
7 **470** builds the application program with the updated item.
8

9 In still another alternative embodiment of the present invention, an application program
10 is updated for execution on the data processing system based upon determining if any items
11 described in the decrypted configuration have been updated. In this alternative embodiment,
12 process block **465** also determines if any items described in the decrypted configuration have
13 been updated. If an item has been updated, then process block **470** retrieves the updated item
14 and builds the application program with the updated item.
15

16 Using the foregoing specification, the invention may be implemented using standard
17 programming and/or engineering techniques using computer programming software, firmware,
18 hardware or any combination or sub-combination thereof. Any such resulting program(s),
19 having computer readable program code means, may be embodied within one or more
20 computer usable media such as fixed (hard) drives, disk, diskettes, optical disks, magnetic tape,
21 semiconductor memories such as Read-Only Memory (ROM), Programmable Read-Only
22 Memory (PROM), etc., or any memory or transmitting device, thereby making a computer
23 program product, i.e., an article of manufacture, according to the invention. The article of
24 manufacture containing the computer programming code may be made and/or used by
25 executing the code directly or indirectly from one medium, by copying the code from one
26 medium to another medium, or by transmitting the code over a network. An apparatus for
27 making, using, or selling the invention may be one or more processing systems including, but
28 not limited to, central processing unit (CPU), memory, storage devices, communication links,
29 communication devices, servers, input/output (I/O) devices, or any sub-components or

individual parts of one or more processing systems, including software, firmware, hardware or any combination or sub-combination thereof, which embody the invention as set forth in the claims. User input may be received from the keyboard, mouse, pen, voice, touch screen, or any other means by which a human can input data to a computer, including through other programs such as application programs, databases, data sets, or files.

One skilled in the art of computer science will easily be able to combine the software created as described with appropriate general purpose or special purpose computer hardware to create a computer system and/or computer sub-components embodying the invention and to create a computer system and/or computer sub-components for carrying out the method of the invention. Although the present invention has been particularly shown and described with reference to a preferred embodiment, it should be apparent that modifications and adaptations to that embodiment may occur to one skilled in the art without departing from the spirit or scope of the present invention as set forth in the following claims.